Documentation Report

Brain Tumor Detection Using YOLO

1. Preprocessing Steps

To prepare the dataset for training and evaluation, the following preprocessing steps were applied:

- **Resizing:** All images were resized to a uniform size of **150 x 150 pixels** to ensure consistency in input dimensions for the model.
- **Normalization:** Pixel values of the images were normalized to a range of **[0, 1]** to improve the model's performance and convergence during training.
- **Data Augmentation:** Data augmentation techniques such as rotation, flipping, and zooming can be employed (although not explicitly shown in the code) to artificially expand the training dataset and improve model generalization.

2. Training Process

The model was trained with the following hyperparameters:

- **Batch Size:** 32
- **Number of Epochs:** 8
- **Early Stopping** to prevent overfitting.
- **Model Checkpointing** to save the best model based on validation accuracy.
- **TensorBoard Logging** for visualizing training progress.

3. Evaluation Metrics

After training, the model was evaluated using the following metrics:

- **Accuracy:** Overall correctness of the model.
- **Precision:** The ratio of true positive predictions to the total predicted positives.
- **Recall (Sensitivity):** The ratio of true positive predictions to the total actual positives.
- **F1-Score:** The harmonic mean of precision and recall, useful for imbalanced classes.

The classification report generated after evaluation provides detailed insights into these metrics for each class.

4. Importing Libraries

- **os:** For interacting with the operating system to manage file paths.
- **numpy:** For numerical operations and array manipulations.

- **matplotlib:** For plotting and visualizing results.
- **tensorflow.keras:** For building and training the deep learning model.
- **ImageDataGenerator:** For generating batches of tensor image data with real-time data augmentation.

5. Instructions to Run the Code

To reproduce the results of this project, follow these steps:

Notes:

1. Dataset Access: Access the dataset via this GitHub link: https://github.com/SartajBhuvaji/Brain-Tumor-Classification-DataSet

2. Running the CNN Model: Run the CNN model using the Brain_Tumor_detection.ipynb file. Open Brain_Tumor_detection.ipynb in Jupyter Notebook and execute the cells.

3. Running the Application Start the application with the main.py file. Use the following command to run the application: uvicorn main:app --reload